EEEEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFFFFF
EEEEEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFFFFF
ÉÉÉÉÉÉÉÉÉÉÉÉÉÉ	RRRRRRRRRRR	FFFFFFFFFFFFF
EEE	RRR RRR	FFF
EEE		
	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
ĒĒĒ	RRR RRR	FFF
EEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEE	RRR RRR	FFF

EEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
	mm mm	111

LP/

LL	000000 00 00 00 00	GGGGGGG GG GG GG GG GG GG GG GG GG GG G	MM MM MMM MMM MMMM MMM MM MM MM MM MM MM	\$	PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
LL		\$			

```
0001
0002
                                   'V04-000'
              Version:
0004
0005
0006
0007
0008
            C *
                 COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
            C *
                  ALL RIGHTS RESERVED.
0009
                 THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0010
0011
0012
0013
0014
                 OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0015
                  TRANSFERRED.
            C ·
0016
0017
            C *
                 THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
            C *
0018
                 AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
            C *
0019
                 CORPORATION.
           C *
0020
           C *
0021
                 DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
           C*
0022
0023
0024
0025
            C *
            C *
           C *
0026
0027
0028
0029
                                                                   Creation date 03-AUG-1983
                       Author Brian Porter
           C
           C
0031
0032
0033
0034
0035
0036
0037
0038
           C + +
                       functional description:
                       Modified by:
                       V03-004 EAD0197
                                                          Elliott A. Drayton
                                                                                                          21-Jun-1984
                                   Changed code to use LSTLUN and RECORD_SIZE from SYECOM.
0040
0041
0042
0043
0044
0045
0046
0047
0051
0053
0055
0055
0055
0055
                       V03-003 SAR0278
                                                                                              21-Jun-1984
                                                          Sharon A. Reynolds
                                   Added TMSCP support.
                                   SAR0171 Sharon A. Reynolds, - Removed the 'b_logmscp' routine.
                                                                                              11-Nov-1983
                       V03-002 SAR0171
                                   - Changed the carriage control for use with ERF.
                                                                                              07-SEP-1983
                       v03-001 BP0001
                                                          Brian Porter,
           C
                                   Added 'no-unit' code.
           ſ
                       Subroutine LOGMS(P (lun,record_length)
                                               'src$:msghdr.for/nolist'
'src$:emblmdef.for/nolist'
                        include
                        include
```

16-Sep-1984 00:05:20 5-Sep-1984 14:00:08

```
include
                                            'src$:syecom.for/nolist'
0312
                      byte
                                           record_length
                      integer * 4
0316
                                           logmscp_type
(emb(16),logmscp_type)
                      integer*2
                      equivalence
0318
0319
0320
                                           mscp_packet(476)
(emb(36),mscp_packet(1))
                      byte
                      equivalence
0321
0323
0323
0324
0325
0326
                                           mscp$$b_opcode
                      integer*4
                                           mscp$$b_endcode
                      integer *4
                      integer*4
                                           mslg$$b_format
                      integer • 4
                                           packet_Tength
                      integer*4
                                           compress4
0328
0329
                      call FRCTOF (Istiun)
0330
                      call HEADER (Istlun)
0331
                      call LOGGER (Istiun, 'ERL$LOGMSCP ENTRY')
0332
                      call LCGMSCP_HEADER ()
0334
0335
                      packet_length = record_size - 36
0336
0337
                      call LINCHK (lstlun,1)
0338
                     write(lstlun,10)
format(' ',:)
0339
0340
           10
0341
0342
0343
                      if (logmscp_type .eq. 9) then
0344
0345
0346
0347
0348
0350
                     mscp$$b_opcode = lib$extzv(0,7,emb(44))
mscp$$b_endcode = lib$extzv(0,8,emb(44))
                      call LINCHK (lstlun,1)
                      write(lstlun,15) 'UNEXPECTED END MESSAGE 'MSCP$W_STATUS''
format(' ',t8,a)
           15
0351
0352
0353
0354
0355
0356
                      call MOVC3 (%val(packet_length),emb(36),emb(38))
call MSCP$B_OPCODE_DISPATCHER (lstlun,mscp$$b_endcode,.true.,.false.)
                      else if (logmscp_type .eq. 10) then
                      mscp$$b_opcode = lib$extzv(0,7,emb(44))
mscp$$b_endcode = lib$extzv(0,8,emb(44))
0358
0359
0360
                      call LINCHK (istlun,1) write(istlun,15) 'UNEXPECTED ATTENTION CONTROL MESSAGE'
0361
0362
                      call MOVC3 (%val(packet_length),emb(36),emb(38))
0364
0365
                      call LINCHK (lstlun,1)
0366
                      write(lstlun,10)
Ď367
```

Page

0396 0397

0398 0399

0400 0401

0402 0403

0404 0405

0406 0407

```
C
C
C
C
```

```
else if (logmscp_type .eq. 11) then
mslg$$b_format = lib$extzv(0,7,emb(44))
call LINCHK (lstlun,1)
if (
  mslg$$b_format .eq. 0
1 .or.
  mslg$$b_format .eq. 1
1) then
write(lstlun,15) 'NO-UNIT DATAGRAM'
write(lstlun_15) 'DATAGRAM FOR NON-EXISTING 'UCB'"
endif
Although the 'embimdef' format is not part of this kind of error log
entry it is necessary, because of the implementation chosen, to zero the equivalent 'emb$b_lm_type' field and re-position the 'mscp' portion
of the entry for the output routines.
call MOVC3 (%val(packet_length),emb(36),emb(38))
emb$b_lm_type = 0
if (mslg$$b_format .eq. 0) then
call MSEG$K_CNT_ERR (lstlun,packet_length)
else if (mslq$$b_format .eq. 1) then
call MSLGSK_BUS_ADDR (lstlun,packet_length)
else if (
  mslg$$b_format .eq. 2
                                   ! Disk transfer error
  OR.
  mslg$$b_format .EQ. 5
                                   ! Tape transfer error
1) then
call DISK_TAPE_TRANSFER_ERRORS (lstlun,packet_length)
  mslg$$b_format .eq. 3
                                   ! SDI error
  .OR.
  mslg$$b_format .eq. 6
                                   ! STI error
```

call DUMPREG (lstlun,(((packet_length - 12) + 3)/4),emb(50))

call MSCP_FIRST_TWELVE_BYTES2 (lstlun,.true.)

if (packet_length .gt. 12) then

call LINCHK (Istlun,1)

write(lstlun,10)

endif

1 .OR.

return

EN

PR

LP

Page

VA

LP

FU

PROGRAM SECTIONS

Name	Bytes	Attributes
SCODE SPDATA SLOCAL SEMB SYECOM	798 208 260 512 44	PIC CON REL LCL SHR EXE RD NOWRT LONG PIC CON REL LCL SHR NOEXE RD NOWRT LONG PIC CON REL LCL NOSHR NOEXE RD WRT LONG PIC OVR REL GBL SHR NOEXE RD WRT LONG PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	1822	

ENTRY POINTS

Address Type Name 0-00000000 LOGMSCP

VARIABLES

Address Type	Name	Address	Type	Name
4-00000012 L*1 4-00000013 L*1 4-00000000 I*4 3-00000000 I*4 3-00000004 I*2 4-0000001E L*1 4-00000010 I*2 4-00000004 I*4 2-00000004 I*4 2-00000004 I*4 2-00000000 I*4 4-00000000 I*4 4-00000000 I*4 4-00000001 L*1	CP-117ZZ DEV CHAR EMB\$B_LM_NAMLNG EMB\$L=HD_SID EMB\$W=HD_ENTRY EMB\$W=LM_MSGTYP END_VALUE FORMS LOGMSCP_TYPE LUN MSCP\$\$B_ENDCODE MSLG\$\$B_FORMAT PACKET_ENGTH RECORD_SIZE VALID_CPU	4-0000011 4-0000014 3-0000010 3-00000015 3-00000012 4-0000010 4-0000001 4-0000017 2-00000000 4-0000008 4-0000008 4-0000019 4-0000018	L*4 L*1 L*1 CH*2 I*2 L*1 L*4 I*4 I*4 L*1 L*1	CP 11780 CRYPTK FLAG EMB\$B_EM_CLASS EMB\$B_LM_TYPE EMB\$T_LM_NAME EMB\$W_HD_ERRSEQ EMB\$W_HM_UNIT EOF FEAG LINES LSTLUN MAILBOX_CHANNEL MSCP\$\$B_OPCODE OPTIONS PRINTER RECORD_LENGTH VALID_CLASS VALID_ENTRY VOLUME_OUTPUT

ARRAYS

Address	Type	Name	Bytes	Dimensions
3-00000000 3-00000026 3-0000006 3-00000024	L+1 L+1 I+4 L+1	EMB EMB\$B_LM_MSGTXT EMB\$Q=HD=TIME MSCP_PACKET	460 8	(0:511) (460) (2) (476)

K 6 16-Sep-1984 00:05:20 5-Sep-1984 14:00:08

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER: [ERF.SRC]LOGMSCP.FOR; 1 Page

LABELS

LOGMSCP

Label Address Address Label Address Label 20' 1-000000A7 1-000000AC 15' 1-000000B3 10'

FUNCTIONS AND SUBROUTINES REFERENCED

Type Name Type Name Type Name DUMPREG I*4 LIB\$EXTZV LOGMSCP_HEADER MSCP_FIRST_TWELVE_BYTES2 MSLG\$K_SML_DSK DISK_TAPE_TRANSFER_ERRORS HEADER COMPRESS4 FRCTOF LINCHK LOGGER MSCP\$B_OPCODE_DISPATCHER MSLG\$K_CNT_ERR MOVC3 MSLGSK BUS ADDR SDI_STI_ERRORS

```
0001
 0002
 0004
 0005
                                               Subroutine LOGMSCP_HEADER ()
 0006
 0007
 0008
                                                include
                                                                                             'src$:msqhdr.for/nolist'
 0067
                                                include
                                                                                              'src$:syecom.for/nolist'
 0195
 0196
                                               byte
                                                                                             class_driver_array(4)
 0197
                                                integer*4
                                                                                            class_driver
 0198
                                                                                             class_driver_string
                                               character*4
 0199
                                                                                             (emb(18),class_driver_array(1),class_driver_string,
                                               equivalence
 0200
                                               1 class_driver)
 0201
 0202
                                                integer *4
                                                                                             cddb$q_cntrlid_array(2)
                                                integer*4
                                                                                             cddb$q_cntrlid1
(cddb$q_cntrlid_array(1),cddb$q_cntrlid1)
 0204
                                               equivalence
 0205
                                                integer*4
                                                                                             cddb$q_cntrlid2
 0206
                                               equivalence
                                                                                             (cddb$\(\bar{q}\)_cntrlid_array(2),cddb$\(\bar{q}\)_cntrlid2)
                                                                                             (emb(22),cddb$q_cntrlid_array(1))
 0207
                                               equivalence
 0208
 0209
                                                                                             cddb$b_systemid_array(6)
                                               byte
 0210
                                                integer *4
                                                                                             cddb$b_systemid1
(cddb$b_systemid_array(1),cddb$b_systemid1)
 0211
                                               equivalence
 0212
                                               integer*2
                                                                                             cddb$b_systemid2
                                                                                             (cddb$b_systemid_array(5),cddb$b_systemid2)
                                               equivalence
 0214
                                                                                             (emb(307,cddb$b_systemid_array(17)
                                               equivalence
 0215
0216
0217
0218
0219
02223
02223
02223
02223
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
02233
0223
                                               byte
                                                                                             fao_array(4)
                                               character*4
                                                                                             fao_string
                                               equivalence
                                                                                             (fao_array(1),fao_string)
                                               external
                                                                                             sys$fao
                                               logical*4
                                                                                             sys$fao
                                               call LINCHK (lstlun,2)
                                               write(lstlun,10) 'CLASS DRIVER',class_driver
                        10
                                               format(/' ',t8,a,t24,z8.8)
                                               if (sys$fao('!AF',,fao_string,%val(4),class_driver_array(1))) then
                                               call LINCHK (istlun.1)
                                               write(lstlun,15) fao_string
format(' ',t40,'/',a,'/')
                        15
                                                endif
                                               call LINCHK (lstlun,2)
                                               write(lstlun,20) cddb$g_cntrlid1,cddb$g_cntrlid2
format(' ',t8,'CDDB$Q_CNTRLID',t24,z8.8,/,
1 t24,z8.8)
                         20
```

EN

LP

PR

7

VA

•

```
16-Sep-1984 05.05:20
5-Sep-1984 14:00:08
LOGMSCP_HEADER
                                                                                                                     VAX-11 FORTRAN V3.4-56
                                                                                                                                                                     Page
                                                                                                                     DISKSVMSMASTER: [ERF.SRC]LOGMSCP.FOR: 1
0243
02445
0246
0247
0248
0251
                     call CDDB$Q_CNTRLID (lstlun,cddb$q_cntlrid1,cddb$q_cntrlid2)
                     call LINCHK (lstlun.2)
                     write(lstlun,25) cddb$b_systemid1,cddb$b_systemid2
format(/' ',t8,'CDDB$B_SYSTEMID',t'4,z8.8,/,
1 t28,z4.4)
           25
0252
0253
0254
                     return
                     end
PROGRAM SECTIONS
                                                                 Attributes
     Name
                                                       Bytes
                                                         242
118
   O SCODE
                                                                 PIC CON REL LCL
                                                                                                             NOWRT LONG
     SPDATA
                                                                 PIC CON REL LCL
                                                                                         SHR NOEXE
                                                                                                             NOWRT LONG
  2 SLOCAL
3 EMB
4 SYECOM
                                                           96
                                                                 PIC CON REL LCL NOSHR NOEXE
                                                                                                         RD
                                                                                                                WRT LONG
                                                         512
                                                                 PIC OVR REL GBL
                                                                                         SHR NOEXE
                                                                                                         RD
                                                                                                                WRT LONG
                                                                 PIC OVR REL GBL
                                                                                         SHR NOEXE
                                                                                                                WRT LONG
     Total Space Allocated
                                                        1012
ENTRY POINTS
     Address Type
                         Name
  0-00000000
                         LOGMSCP_HEADER
VARIABLES
     Address Type
                         Name
                                                                         Address Type Name
                        CDDB$B_SYSTEMID1
CDDB$Q_CNTLRID1
CDDB$Q_CNTRLID2
CLASS_DRIVER_STRING
CP_11780
CRYPTK_FLAG
                                                                                             CDDB$B_SYSTEMID2
CDDB$Q_CNTRLID1
   3-0000001E
                  1+4
                                                                       3-00000022
   2-000000004
                  R+4
                                                                       3-00000016
                                                                                       1+4
                                                                      3-00000012
4-00000013
4-00000013
                                                                                             CLASS DRIVER
CP_11750
CP_11722
   3-0000001A
                   1+4
                                                                                       I +4
  3-00000012
4-00000011
                  CHAR
                                                                                      L + 1
                  L+1"
                                                                                      L+1
                                                                                             DEV_CHAR
   4-00000014
                   L . 4
                                                                       4-000000D
                                                                                       I ±4
                         EMB$L_RD_SID
EMB$W_HD_ERRSEQ
   3-00000000
3-0000000E
                   I +4
                                                                       3-00000004
                                                                                       1 * 2
                                                                                             EMB$W_HD_ENTRY
                                                                                      L+1 END_VALUE
CHAR FAO_STRING
                   1+2
                                                                       4-0000001E
                                                                                      L * 1
                  L+1
   4-0000001D
                         EOF_FEAG
                                                                       2-00000000
                  L+4
   4-00000004
                         FORMS
                                                                       4-0000000C
                                                                                       L+1
                                                                                             LINES
                                                                                      1+4
   4-00000027
                   Ī+4
                         LSTLUN
                                                                       4-0000001F
                                                                                             MAILBOX_CHANNEL
   4-0000002B
                   CHAR OPTIONS
                                                                       4-00000008
                                                                                       L +4
                                                                                             PRINTER'
   4-00000000
                   1+4
                         RECENT
                                                                       4-00000023
                                                                                             RECORD SIZE VALID CPU
                                                                                       I +4
```

L+1

VALID_TYPE

4-0000001A

4-0000001C L*1

L+1

L +1

L+1

4-00000019 4-00000018 4-00000018 VALID_CLASS VALID_ENTRY

VOLUME_OUTPUT

LP

A

AR

LA

FU

COI

CO

1-00000053 25'

FUNCTIONS AND SUBROUTINES REFERENCED

Type Name Type Name Type Name

CDDB\$Q_CNTRLID

LINCHK

1-00000032 20'

L*4 SYS\$FAO

COMMAND QUALIFIERS

1-00000018 10'

FORTRAN /LIS=LIS\$:LOGMSCP/OBJ=OBJ\$:LOGMSCP MSRC\$:LOGMSCP

1-00000025 15'

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM) /SHOW=(NOPREPROCESSOR, NOINCLODE, MAP) /F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

COMPILATION STATISTICS

Run Time: 4.86 seconds 14.09 seconds 187 Elapsed Time: Page Faults: 183 pages Dynamic Memory:

EQUIPMENT CORPORATION AH-BT13A-SE DIGITAL AND PROPRIETARY CONFIDENTIAL VAX/VMS V4.0 interior IE IE The It THE RESERVE OF THE PARTY OF THE DATE OF THE PARTY A. Maria IK I In th